What is claimed is:

1. A method for dispatching requests to processing resources, the method comprising steps of:

determining if a processing resource is idle, the processing resource having a current service type to process requests that have the current service type;

determining if the processing resource is to be switched to a different service type to process requests having the different service type when the processing resource is idle;

switching the processing resource to the different service type when the switching is determined; and

dispatching an outstanding request having the different service type to the processing resource.

2. The method as claimed in claim 1, wherein the switch determining step comprises the steps of:

determining if there is an outstanding request having the current service type; and

identifying a service type of a currently outstanding request when there is no outstanding request having the current service type; and

determining that the processing resource is to be switched to the identified service type.

3. The method as claimed in claim 2, wherein the switch determining step determines not to switch the processing resource when a request having the current service type is expected to arrive in a shorter period than a period for switching the processing resource to the identified service type.

4. The method as claimed in claim 1, wherein a service type is defined by a primary request parameter and one or more secondary request parameters, and the switching step switches the processing resource to the different service type that has a same primary request parameter as the current service type.

5. The method as claimed in claim 4 further comprising a step of queuing requests in a plurality of queues, each queue being used for queuing requests having a same primary request parameter.

6. The method as claimed in claim 5, wherein the switch determining step comprises steps of:

determining if there is a queued request having the current service type in a queue; and

identifying a service type of a currently queued request when there is no queued request having the current service type in the queue; and

determining that the processing resource is to be switched to the identified service type.

7. The method as claimed in claim 6, wherein the identifying step identifies a service type of a first queued request which is the head of the queue.

8. The method as claimed in claim 6, wherein the switch determining step determines not to switch the processing resource when a request having the current service type is expected to arrive in a shorter period than a period for switching the processing resource to the identified service type.

9. The method as claimed in claim 6, wherein the switch determining step determines if the server instance is to be switched by invoking a balancing algorithm using preparation costs for switching the processing resource to the identified service type.

10. The method as claimed in claim 1 further comprising a step of allowing dispatching of an outstanding request having the current service type from a queue prior to one or more outstanding requests that have a different service type and arrived at the queue before the outstanding request having the current service type.

11. The method as claimed in claim 1 further comprising a step of terminating the processing resource if the processing resource is determined not to switched and it is idle for longer than a predetermined time period.

12. A method for dispatching queued requests to a predetermined number of server instances, the method comprising steps of:

determining if a server instance is idle, the server instance having a current service type to process requests that have the current service type;

determining if the server instance is to be switched to a different service type to process requests having the different service type when the server instance is idle;

switching the server instance to the different service type when the switching is determined; and

dispatching a queued request having the different service type to the server instance.

13. The method as claimed in claim 12, wherein a service type is defined by a primary request parameter and one or more secondary request parameters, and requests are queued in a plurality of queues, each queue being used for queuing requests having a same primary request parameter; and

the switch determining step comprises the steps of:

determining, for a queue, if there is a queued request having the current service type; and

identifying a service type of a currently queued request when there is no queued request having the current service type; and

determining if the server instance is to be switched based on the identified service type.

14. The method as claimed in claim 13, wherein the identifying step identifies a service type of a first queued request which is the head of the queue.

15. The method as claimed in claim 13, wherein the switch determining step determines not to switch the server instance when a request having the current service type is expected to arrive in a shorter period than a period for switching the server instance to the identified service type.

16. The method as claimed in claim 13, wherein the switch determining step determines if the server instance is to be switched by invoking a balancing algorithm using preparation costs for switching the server instance to the identified service type.

17. The method as claimed in claim 12 further comprising a step of allowing dispatching of a queued request having the current service type from a queue prior to one or more queued requests that have a different service type and arrived at the queue before the queued request having the current service type.

18. The method as claimed in claim 12 further comprising a step of terminating the server

instance if the server instance is determined not to switched and it is idle for longer than a predetermined time period.

19. The method as claimed in claim 12 further comprising steps of:

reserving a minimum number of server instance slots for each queue, each server instance slot representing a potential server instance; and

allocating one or more non-reserved server instance slots for one or more queues when the total number of server instances is larger than the sum of minimum numbers of reserved server instance slots for queues being used.

20. The method as claimed in claim 19 further comprising a step of:

reallocating a non-reserved server instance slot to a different queue when the non-reserved server instance slot is free.

21. The method as claimed in claim 20, wherein the reallocating step comprises steps of:

selecting a queue having fewest allocated non-reserved server instance slots; and

reallocating the non-reserved server instance slot to the selected queue.

22. The method as claimed in claim 21, wherein primary request parameters of service types relate to priority, and the selecting step selects a higher queue having a higher priority primary request parameter if there are multiple queues having the fewest allocated non-reserved server instance slots.

23. The method as claimed in claim 21, wherein the selecting step comprises steps of:

checking if there are at least the minimum number of server instances running requests at the selected queue; and

selecting a next queue having next fewest allocated non-reserved server instance slots when there are at least the minimum number of server instances running requests at the selected queue.

24. The method as claimed in claim 22, wherein the selecting step comprises steps of:

checking if there are at least the minimum number of server instances running requests at the selected queue; and

selecting a higher queue having a higher priority to allow borrowing of a server instance by the higher queue.

25. A method for dispatching queued requests to a predetermined number of server instances, the method comprising steps of:

using a plurality of queues for queuing requests, each request having a service type, a service type being defined by a primary request parameter and one or more secondary request parameters, and each queue being used for queuing requests having a same primary request parameter;

reserving a minimum number of server instance slots for each queue, each server instance slot representing a potential server instance, each server instance having a current service type;

allocating one or more non-reserved server instance slots for one or more queues when the total number of server instances is larger than the sum of minimum numbers of reserved server instance slots for queues being used;

reallocating a non-reserved server instance slot to a different queue when the non-reserved server instance slot is free; and

dispatching a queued request from a queue to an idle server instance in a server instance slot allocated for the queue.

26. The method as claimed in claim 25, wherein the reallocating step comprises steps of:

selecting a queue having fewest allocated non-reserved server instance slots; and

reallocating the non-reserved server instance slot to the selected queue.

27. The method as claimed in claim 26, wherein primary request parameters of service types relate to priority, and the selecting step selects a higher queue having a higher priority primary request parameter if there are multiple queues having the fewest allocated non-reserved server instance slots.

28. The method as claimed in claim 26, wherein the selecting step comprises steps of:

checking if there are at least the minimum number of server instances running requests at the selected queue; and

selecting a next queue having next fewest allocated non-reserved server instance slots

when there are at least the minimum number of server instances running requests at the selected queue.

29. The method as claimed in claim 27, wherein the selecting step comprises steps of:

checking if there are at least the minimum number of server instances running requests at the selected queue; and

selecting a higher queue having a higher priority to allow borrowing of a server instance by the higher queue.

30. A request dispatching system for dispatching requests to processing resources, the request dispatching system comprising:

a processing resource controller having a switch controller for controlling switching of an idle processing resource having a current service type to a different service type; and

a dispatching controller for dispatching an outstanding request having the different service type to the processing resource.

31. The request dispatching system as claimed in claim 30, wherein the switch controller comprises:

a request searcher for searching an outstanding request having the current service type; and

an identifier for identifying a service type of a currently outstanding request to switch the processing resource to the identified service type.

32. The request dispatching system as claimed in claim 31, wherein the switch controller has a comparator for comparing a expected period for a request having the current service type to arrive and a switching period for switching the processing resource to the identified service type to switch the processing resource when the expected period is longer than the switching period.

33. The request dispatching system as claimed in claim 30, wherein a service type is defined by a primary request parameter and one or more secondary request parameters, and the switching controller switches the processing resource to the different service type that has a same primary request parameter as the current service type.

34. A request dispatching system for dispatching queued requests to a predetermined number of server instances, the request dispatching system comprising:

a server instance controller having a switch controller for controlling switching of an idle server instance having a current service type to a different service type; and

a dispatching controller for dispatching an outstanding request having the different service type to the server instance..

35. The request dispatching system as claimed in claim 34, wherein the switch controller comprises:

a request searcher for searching an queued request having the current service type; and

an identifier for identifying a service type of a currently queued request to switch the server instance to the identified service type.

36. The request dispatching system as claimed in claim 35, wherein the switch controller has a comparator for comparing a expected period for a request having the current service type to arrive and a switching period for switching the server instance to the identified service type to switch the server instance when the expected period is longer than the switching period.

37. The request dispatching system as claimed in claim 34, wherein a service type is defined by a primary request parameter and one or more secondary request parameters, and the switching controller switches the server instance to the different service type that has a same primary request parameter as the current service type.

38. The request dispatching system as claimed in claim 34 further comprising a skip controller for allowing dispatching of a queued request having the current service type from a queue prior to one or more queued requests that have a different service type and arrived at the queue before the queued request having the current service type.

39. The request dispatching system as claimed in claim 34 further comprising an allocation controller for reserving a minimum number of server instance slots for each queue, each server instance slot representing a potential server instance; allocating one or more non-reserved server instance slots for one or more queues when the total number of server

instances is larger than the sum of minimum numbers of reserved server instance slots for queues being used, and reallocating a non-reserved server instance slot to a different queue when the non-reserved server instance slot is free.

40. The request dispatching system as claimed in claim 34, wherein the allocation controller comprises a selector for selecting a queue having fewest allocated non-reserved server instance slots to reallocate the non-reserved server instance slot to the selected queue.

41. A computer readable memory for storing computer executable instructions for carrying out a method for dispatching requests to processing resources, the method comprising steps of:

determining if a processing resource is idle, the processing resource having a current service type to process requests that have the current service type;

determining if the processing resource is to be switched to a different service type to process requests having the different service type when the processing resource is idle;

switching the processing resource to the different service type when the switching is determined; and

dispatching an outstanding request having the different service type to the processing resource.

42. A computer readable memory for storing computer executable instructions for carrying out a method for dispatching queued requests to a predetermined number of server instances, the method comprising steps of:

using a plurality of queues for queuing requests, each request having a service type, a service type being defined by a primary request parameter and one or more secondary request parameters, and each queue being used for queuing requests having a same primary request parameter;

reserving a minimum number of server instance slots for each queue, each server instance slot representing a potential server instance, each server instance having a current service type;

allocating one or more non-reserved server instance slots for one or more queues when the total number of server instances is larger than the sum of minimum numbers of reserved server instance slots for queues being used;

reallocating a non-reserved server instance slot to a different queue when the non-

reserved server instance slot is free; and

dispatching a queued request from a queue to an idle server instance in a server instance slot allocated for the queue.

43. Electronic signals for use in the execution in a computer of a method for dispatching requests to processing resources, the method comprising steps of:

determining if a processing resource is idle, the processing resource having a current service type to process requests that have the current service type;

determining if the processing resource is to be switched to a different service type to process requests having the different service type when the processing resource is idle;

switching the processing resource to the different service type when the switching is determined; and

dispatching an outstanding request having the different service type to the processing resource.

44. Electronic signals for use in the execution in a computer of a method for dispatching queued requests to a predetermined number of server instances, the method comprising steps of:

using a plurality of queues for queuing requests, each request having a service type, a service type being defined by a primary request parameter and one or more secondary request parameters, and each queue being used for queuing requests having a same primary request parameter;

reserving a minimum number of server instance slots for each queue, each server instance slot representing a potential server instance, each server instance having a current service type;

allocating one or more non-reserved server instance slots for one or more queues when the total number of server instances is larger than the sum of minimum numbers of reserved server instance slots for queues being used;

reallocating a non-reserved server instance slot to a different queue when the non-reserved server instance slot is free; and

dispatching a queued request from a queue to an idle server instance in a server instance slot allocated for the queue.